Benchmark: Neural Network Malware Classification

Preston K. Robinette, Diego Manzanas Lopez, and Taylor T. Johnson

Vanderbilt University, Nashville TN 37212, USA

Abstract. As malware threats continue to increase in both complexity and sophistication, the adoption of advanced detection methods, such as deep neural networks (DNNs) for malware classification, has become increasingly vital to safeguard digital infrastructure and protect sensitive data. In order to measure progress in this safety-critical landscape, we propose two malware classification benchmarks: a feature-based benchmark and an image-based benchmark. Feature-based datasets provide a detailed understanding of malware characteristics, and image-based datasets transform raw malware binary data into grayscale images for swift processing. These datasets can be used for both binary classification (benign vs. malicious) as well as classifying known malware into a particular family. This paper, therefore, introduces two benchmark datasets for binary and family classification with varying difficulty levels to quantify improvements in malware classification strategies. Key contributions include the creation of feature and image dataset benchmarks, and the validation of a trained binary classification network using the feature dataset benchmark. Benchmarks as well as example training code are available[1].

Keywords: Malware · Verification · Benchmarks

1 Introduction

Malware is any software designed with malicious intent. Various types of harmful software include stealing sensitive data (data theft [6, 4]), monitoring activity and passwords (espionage [15]), the creation of a 'botnet' (Distributed Denial of Service (DDoS) [1]), stealing data and holding it for ransom (ransomware attacks [7]), using someone else's system to mine cryptocurrency (cryptojacking [11]), and distributing spam [5]. To mitigate the harmful effects of malware, defense mechanisms rely upon the ability to distinguish between benign software and malicious software. If malware can be detected, it can be effectively isolated and neutralized before it has a chance to compromise the system. This early detection is not only pivotal for securing individual systems but also plays a crucial role in preventing the spread of malware across networks, thereby fortifying overall cybersecurity infrastructure. It also facilitates the process of reverse

[1] Code: https://github.com/pkrobinette/malware_benchmarks

engineering the malware to understand its functionality and to devise robust countermeasures for similar threats in the future.

As the sophistication of these threats continues to increase, more advanced methods for detection and classification have become necessary. One such method is the application of deep neural networks (DNN) for malware classification [8, 13, 3, 12, 10, 2]. Neural network malware classifiers are usually trained on either feature datasets or image datasets. Feature datasets consist of numerical or categorical attributes extracted from malware samples, such as opcode frequencies, system calls, file sizes, and other static or dynamic features. These feature sets are designed to capture the distinctive behaviors or properties of malicious software. On the other hand, image datasets involve a transformation of the raw binary data of malware files into grayscale images, which depict the visual patterns underlying the malware's binary structure.

Both approaches present their own advantages. Feature-based datasets can provide a detailed overview of the malware's characteristics, facilitating a deeper understanding of the malware's functionality and behavior. This approach, however, can be time-consuming and requires expert knowledge to identify and extract the most relevant features. In contrast, image-based datasets can be processed quickly and do not necessitate the tedious task of feature selection.

While feature datasets are most commonly used for binary classification of either benign or malicious, image datasets tend to also be used to classify malware families. This is particularly useful, as malware belonging to the same family often share a common code base and exhibit similar patterns of behavior. By identifying the malware family, cybersecurity experts can gain insights into the malware's likely origin, its potential behavior, and the most effective countermeasures. Additionally, this classification aids in tracking the evolution of different malware strains and anticipates emerging threats.

As threats continue to evolve and the methods we use to combat against them improve as well, it is imperative to provide benchmark datasets in this domain to quantify improvements in both classification strategies as well as verification tools. In addition to datasets utilized in this paper, which are representative of the state-of-the-art in this domain, the property verified is also an important factor as well. As this is the first benchmark for malware classification, the verification process needs to be thorough and comprehensive, ensuring that the system's robustness is not solely reliant on specific data but can generalize well across varied and potentially unseen malicious threats. Thus, this paper introduces two benchmark datasets (a feature dataset and an image dataset), which evaluate how robust a classifier is against adversarial attacks and dataset shifts. The contributions of this work, therefore, are the following:

1. Introduction of a feature dataset benchmark for binary classification, consisting of three different difficulty levels.
2. Introduction of an image dataset benchmark for family classification, consisting of three different difficulty levels.
3. A trained binary classification network verified using the feature dataset benchmark.

2 Preliminaries

In this section, we introduce robustness and the malware datasets used for benchmarking: (1) a feature dataset, which represents malware samples as vectors of data such as byte entropy and string length, and (2) an image dataset, which represents malware samples as grayscale images.

2.1 Feature Datasets

Malware feature datasets are composed of 'features' extracted from collected samples, which can be either benign or malicious. Common features include file properties, binary content, API calls, network activities, registry key modifications, and embedded resources. These data points form a comprehensive profile of each software sample, providing vital clues about its behavior, origin, and potential harm. By analyzing these features, cybersecurity experts can accurately classify software as benign or malicious and identify novel malware variants. In addition to static features, feature datasets can include dynamic attributes, such as runtime behavior, system interactions, and state changes over time. Dynamic attributes provide insights into how the software behaves when executed, including changes made to files, registries, and the system memory. They can also capture network connections initiated, services used, and any suspicious activities like process injection, encryption of user files, or attempts to evade detection. These dynamic behaviors, combined with static features, help create a more holistic understanding of the malware's functionality and impact, enhancing the effectiveness of threat detection and prevention mechanisms.

BODMAS While there are many publicly available feature datasets, we utilize the Blue Hexagon Open Dataset for Malware Analysis, or BODMAS [14]. BODMAS contains 77,142 benign samples (marked as label 0) and 57,293 malicious samples (marked as label 1). Each sample is represented by a vector of 2381 features extracted using both dynamic and static analysis methods with the aid of the LIEF project [9]. Table 1 shows the seven distinct datatype categories contained in the dataset. These data types are used to distinguish between benchmark difficulty levels.

2.2 Image Datasets

As extracting static and dynamic features requires expert knowledge and is time-intensive, researchers have also utilized image alternatives. A sample binary is segmented into bytes, and these bytes are then converted into grayscale pixel values. For instance, if the binary file contains the sequence 0100010010010111, this would be chunked into 8 bits (a byte): [01000100, 10010111], and converted to decimal: [68, 151]. These would be two-pixel values in the corresponding grayscale image of the binary. Image-level representations provide a quick alternative to static and dynamic analysis, while still providing valuable insights into the structure and patterns inherent in the binary data. The graphical patterns formed by these binary sequences can be distinctive, enabling models to learn

Table 1. The data types contained in the BODMAS dataset. Each feature data type has a distinct range, which demonstrates the need for a range-specific perturbation.

Feature Type	Count	Max Range Pre-Scale	Max Range Post-Scale	Example
Continuous	5	[5.0, 2.0e5]	[-0.1, 304.6]	Entropy of the file
Categorical	8	[0.0, 6.5e4]	[-0.0, 124.3]	Machine type
Hash Categorical	560	[-647.6, 15.4]	[0.0, 361.0]	Hash of original file
Discrete with Large Range	34	[0.0, 4.3e9]	[-0.0, 261.6]	Number of occurrences of each byte value within the file
Binary	5	[0.0, 1.0]	[-2.1, 0.5]	Presence of debug section
Hash Categorical Discrete	1531	[-8.0e6, 1.6e9]	[-327.9, 164.0]	Hash of target system type
Memory Related	16	[0.0, 4.0e9]	[-0.1, 307.5]	Size of the original file
Null	222	[-31.0, 60.0]	[-0.9, 160.4]	—

and identify specific behaviors and characteristics of the represented software. This approach significantly reduces the time and complexity of feature extraction while maintaining a high level of analysis precision.

Malimg We utilize the Malimg Dataset in this work. The Malimg dataset is composed of 9339 malware images from 25 different malware families [8]. Table 2 shows the breakdown of samples per family as well as the family hierarchies. For instance, *Allaple.A* and *Allaple.L* are two different families that fall under the same type of attack: *Worm*. As there are large differences between the number of represented samples (*Allaple.A*: 2949 samples vs. *Skintrim.N*: 80 samples), this adds to the difficulty of verifying specific malware families.

2.3 Robustness

As malware attacks evolve, we want classification systems that are robust against changing adversarial attacks. To quantify this goal, these benchmarks focus on a robustness verification property – given changes in a sample (changing adversarial attacks), can the classification system correctly classify samples even in the presence of change? For instance, let a neural network classifier be denoted as f, an input $x \in \mathbb{R}^{n \times m}$, target $y \in \mathbb{R}^N$ where N is the number of classes, perturbation parameter $\epsilon \in \mathbb{R}$, and an input set R containing x_p such that $X_p = \{x : ||x - x_p||_\infty \leq \epsilon\}$ which represents the set of all possible perturbations of x where $||x - x_p||_\infty$ is the $\mathcal{L}_\infty$ norm. A model is robust at x if all the perturbed inputs x_p are classified to the same label as y, e.g., the system is **robust** if $f(x_p) = f(x) = y$ for all $x_p \in X_p$. In this way, we can verify that a classification system will be useful in future situations, not just against known malware patterns in the present.

Table 2. Malimg dataset details.

No.	Family	Family Name	No. of Samples
01	Dialer	Adialer.C	122
02	Backdoor	Agent.FYI	116
03	Worm	Allaple.A	2949
04	Worm	Allaple.L	1591
05	Trojan	Alueron.gen!J	198
06	Worm:AutoIT	Autorun.K	106
07	Trojan	C2Lop.P	146
08	Trojan	C2Lop.gen!G	200
09	Dialer	Dialplatform.B	177
10	Trojan Downloader	Dontovo.A	162
11	Rogue	Fakerean	381
12	Dialer	Instantaccess	431
13	PWS	Lolyda.AA 1	213
14	PWS	Lolyda.AA 2	184
15	PWS	Lolyda.AA 3	123
16	PWS	Lolyda.AT	159
17	Trojan	Malex.gen !J	136
18	Trojan Downloader	Obfuscator.AD	142
19	Backdoor	Rbot!gen	158
20	Trojan	Skintrim.N	80
21	Trojan Downloader	Swizzor.gen!E	128
22	Trojan Downloader	Swizzor.gen!I	132
23	Worm	VB.AT	408
24	Trojan Downloader	Wintrim.BX	97
25	Worm	Yuner.A	800

3 Benchmarks

In this section, we introduce each of the malware benchmarks in more detail.

3.1 Malware Feature Benchmark

The malware feature benchmark consists of 200 samples taken from a stratified sampling of the entire BODMAS dataset, giving a split of 43% malicious samples. While the specific samples do not have varying levels of difficulty, the data type used during the perturbation as well as the size of the perturbation lends itself to levels of verification difficulty. The level, corresponding data type, and epsilon used for each benchmark are described in Table 3.

The $\mathcal{L}_\infty$ perturbation of size ϵ^* is applied to the range of each feature of the corresponding datatype. For example, if feature 1 has a range $[3, 567]$, the $\mathcal{L}_\infty$ bound with $\epsilon^* = 0.1\%$ would be $\pm 0.56 = 0.1\% \times (567 - 3)$. ϵ^* here represents the modification to the range of each feature. For all samples s and for the designated features f (dependent on benchmark level) within that sample, the

Table 3. Malware feature dataset details (BODMAS).

Benchmark Level	Perturbation Data Type	Perturbation Size(ϵ^*)
Level 1	Continuous	0.01
Level 2	Continuous and Discrete	0.025
Level 3	All	0.001

Table 4. Malware image dataset details (Malimg).

Benchmark Level	Perturbation Size(ϵ)
Level 1	5
Level 2	10
Level 3	15

perturbation ϵ used for verification is defined as ϵ^* of the range of that feature, i.e., $\epsilon_{s,f} = \epsilon^* \text{range}_{s,f}$. This provides a more feature realistic perturbation to each sample. Even though the perturbed features will all be of the same datatype, the ranges for those features can be drastically different, making it important to consider the range for each particular feature of the corresponding datatype.

3.2 Malware Image Benchmark

Whereas the malware feature dataset verifies different data types and perturbation sizes, the malware image dataset only focuses on verifying different perturbation sizes in each level. The verification dataset consists of 5 randomly sampled images from each malware family, which makes a total of 125 images of pixel values in the range $[0, 255]$. Each level consists of an $\mathcal{L}_\infty$ perturbation on the pixel values of the verification image. A perturbation size of $\epsilon = 3$ indicates that the value of each pixel in the image is allowed to be changed by ± 3. The level and corresponding perturbation sizes ϵ are shown in Table 4.

4 Benchmark Demonstration

To demonstrate the use of these benchmarks, we demonstrate the verification of a binary neural network classifier trained on the BODMAS dataset. First, we train a neural network with an input layer of 2381, one hidden layer of size 32, and an output layer of size 2 (binary classification). This model is trained with an Adam optimizer and a learning rate of 0.001 for 20 epochs. The testing performance of this trained model is shown in Table 5.

From these results, the model performs well on the test set, as indicated by the high value for each metric. We then verify this model using the level 2 feature benchmark, which results in the successful robustness verification of 103 out of 200 images[2]. This means that only about 50% of the tested samples were verified

[2] Code available here: https://github.com/pkrobinette/malware_benchmarks

Table 5. Binary neural network classifier model performance.

Metric	**Value**
Accuracy	1.0
Precision	0.99
Recall	1.0
F1	1.0

within the given perturbation; this does not bode well for the tested classifier as we would want a higher verification rate to ensure confidence in its robustness, especially in real-world scenarios where diverse adversarial attacks might be more prevalent. This result highlights the importance of using benchmarks to compare trained classifiers. Just based on the metrics, the trained model performs well. Considering the verification, however, our model may not be as robust as we would like.

5 Conclusion

In this work, we introduce two malware classifier benchmarks: a feature benchmark and an image benchmark. Each benchmark consists of 3 different difficulty levels. The feature benchmark is distinguished by the data types perturbed, whereas the image benchmark is distinguished by perturbation size. These novel benchmarks will be pivotal in quantifying improvements in the safety-critical domain of malware classification.

Acknowledgements This paper was supported in part by a fellowship award under contract FA9550-21-F-0003 through the National Defense Science and Engineering Graduate (NDSEG) Fellowship Program, sponsored by the Air Force Research Laboratory (AFRL), the Office of Naval Research (ONR), and the Army Research Office (ARO). The material presented in this paper is based upon work supported by the National Science Foundation (NSF) through grant numbers 2220426 and 2220401, the Defense Advanced Research Projects Agency (DARPA) under contract number FA8750-23-C-0518, and the Air Force Office of Scientific Research (AFOSR) under contract number FA9550-22-1-0019 and FA9550-23-1-0135. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of AFOSR, DARPA, or NSF.

References

1. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., et al.: Understanding the mirai botnet. In: 26th {USENIX} security symposium ({USENIX} Security 17). pp. 1093–1110 (2017)

2. Awan, M.J., Masood, O.A., Mohammed, M.A., Yasin, A., Zain, A.M., Damaševičius, R., Abdulkareem, K.H.: Image-based malware classification using vgg19 network and spatial convolutional attention. Electronics **10**(19), 2444 (2021)
3. Bhodia, N., Prajapati, P., Di Troia, F., Stamp, M.: Transfer learning for image-based malware classification. arXiv preprint arXiv:1903.11551 (2019)
4. Carvalho, M., DeMott, J., Ford, R., Wheeler, D.A.: Heartbleed 101. IEEE Security & Privacy **12**(4), 63–67 (2014). https://doi.org/10.1109/MSP.2014.66
5. Khan, I., Kwon, Y.W.: Attention-based malware detection of android applications. In: 2022 IEEE International Conference on Big Data (Big Data). pp. 6693–6695 (2022). https://doi.org/10.1109/BigData55660.2022.10020684
6. Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Fogh, A., Horn, J., Mangard, S., Kocher, P., Genkin, D., Yarom, Y., Hamburg, M.: Meltdown: Reading kernel memory from user space. In: 27th USENIX Security Symposium (USENIX Security 18) (2018)
7. McIntosh, T., Kayes, A.S.M., Chen, Y.P.P., Ng, A., Watters, P.: Ransomware mitigation in the modern era: A comprehensive review, research challenges, and future directions. ACM Comput. Surv. **54**(9) (oct 2021). https://doi.org/10.1145/3479393, https://doi.org/10.1145/3479393
8. Nataraj, L., Karthikeyan, S., Jacob, G., Manjunath, B.S.: Malware images: visualization and automatic classification. In: Proceedings of the 8th international symposium on visualization for cyber security. pp. 1–7 (2011)
9. Oyama, Y., Miyashita, T., Kokubo, H.: Identifying useful features for malware detection in the ember dataset. In: 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW). pp. 360–366 (2019). https://doi.org/10.1109/CANDARW.2019.00069
10. Singh, A., Handa, A., Kumar, N., Shukla, S.K.: Malware classification using image representation. In: Cyber Security Cryptography and Machine Learning: Third International Symposium, CSCML 2019, Beer-Sheva, Israel, June 27–28, 2019, Proceedings 3. pp. 75–92. Springer (2019)
11. Tekiner, E., Acar, A., Uluagac, A.S., Kirda, E., Selcuk, A.A.: Sok: cryptojacking malware. In: 2021 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 120–139. IEEE (2021)
12. Vasan, D., Alazab, M., Wassan, S., Naeem, H., Safaei, B., Zheng, Q.: Imcfn: Image-based malware classification using fine-tuned convolutional neural network architecture. Computer Networks **171**, 107138 (2020)
13. Vasan, D., Alazab, M., Wassan, S., Safaei, B., Zheng, Q.: Image-based malware classification using ensemble of cnn architectures (imcec). Computers & Security **92**, 101748 (2020)
14. Yang, L., Ciptadi, A., Laziuk, I., Ahmadzadeh, A., Wang, G.: Bodmas: An open dataset for learning based temporal analysis of pe malware. In: 4th Deep Learning and Security Workshop (2021)
15. Zhang, F., Wang, H., Leach, K., Stavrou, A.: A framework to secure peripherals at runtime. In: Kutyłowski, M., Vaidya, J. (eds.) Computer Security - ESORICS 2014. pp. 219–238. Springer International Publishing, Cham (2014)